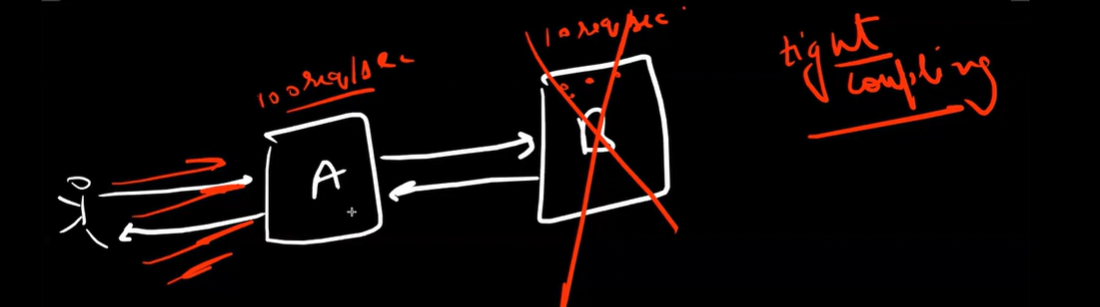
**Tasks To Be Performed:**

1. Create a FIFO SQS queue and test by sending messages.

2. Register your mail in SES and send a test mail to yourself.

SQS(Amazon Simple Queue Service)🡪Let’s we have a microservice framework where A application is tightly coupled with the B Application means A is directly talking to B.



Now A is fast taking 100 requests/sec and B is slow taking only 10 requests and all are synchronous at some time there are lots of requests pile up on B and it may crash.

So microservice always prefer loose coupling and put a queue between two frameworks.

Now what does this queue do?

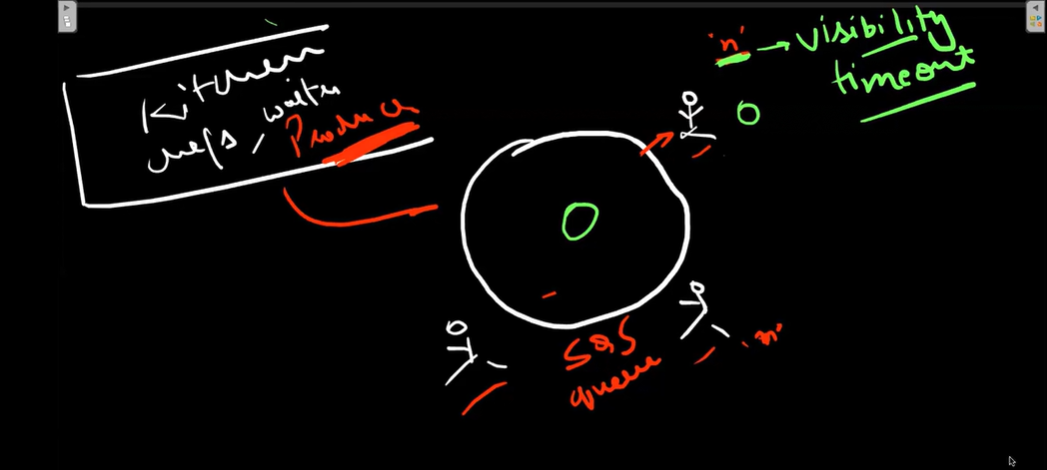
It is like a temporary storage , all the requests from A will be accumulated here and B take request from the queue. It is like a buffer.

So, if B crashes all the requests which will come from A will save in queue and when B is up it will start taking request from there . so how this it is useful. No System fail will happen.

There is many soln. in market like Kafka . Aws offers SQS. It is Serverless.

It is based on producer/consumer algorithm. It is pull based and message is consumed by one consumer.

Producer is keeping sending message and consumer use to consume the message in n number of seconds( known as visibility timeout) otherwise it will be sent back again to the queue.



Now let’s create a sqs🡪

Search simple queue service🡪 create queue

A screenshot of a computer

Description automatically generated

Here standard for unordered and FIFO is for ordered queue.

Give any name.

A white background with black and orange text

Description automatically generated

Visibility timeout sets the length of time that a message received from a queue (by one consumer) will not be visible to the other message consumers.

The visibility timeout begins when Amazon SQS returns a message. If the consumer fails to process and delete the message before the visibility timeout expires, the message becomes visible to other consumers. If a message must be received only once, your consumer must delete it within the duration of the visibility timeout.

Here retention period is the time for a message to be available in queue . if due to any condition message remain in queue it will not delete it till this period.

If your consumers need additional time to process messages, you can delay each new message coming to the queue. The delivery delay is the amount of time to delay the first delivery of each message added to the queue. Any messages that you send to the queue remain invisible to consumers for the duration of the delay period.

256 KB is the maximum message size we use to send but we can increase it by code in java.

Now main receive message wait time , as we know consumer use to poll the queue for message in every interval . this is for the long polling like a consumer send a request it will wait for given time till it got the message to process . which will decrease number of request as it is serverless so it will also reduce cost.

We will see below options when we are doing demo.

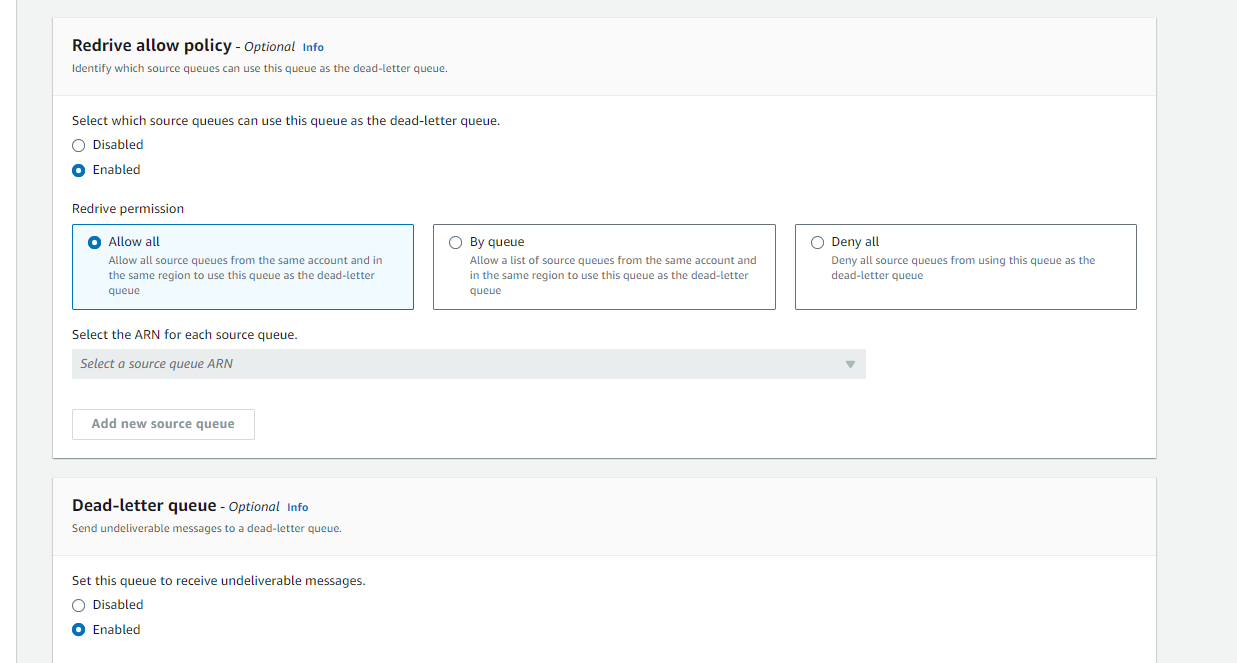
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Here we can define who can send and read the message from the queue.



Here we can define another queue which will work as dead letter queue means if due to any reason message fail to deliver in our queue . it will send it to this queue from where we can retrieve it also which is known as redrive.

Now create queue. A screenshot of a computer

Description automatically generated

We can see all the details of our queue here.

Default visibility timeout is 15 sec here.

Now lets send and receive message. Click on send and receive message.

A screenshot of a computer

Description automatically generated

Just type your message here.

Message group id to uniquely identify your message related to a particular task.

Now if we send message, we can clearly see it in the queue , but what happen if we send the same message again. It will not go queue will not allow the message with same body again for the 15 mins which is known as deduplication interval for this we have to add a duplication id here.

If u add and send message, we can see count increase for the messages.

A close-up of a white background

Description automatically generated

Remember to send every message with the same body every time we have to add a id.

Now our polling rate is 30 and availability time is 15 sec means a consumer will take 15 sec to consume the messages else it will return to the queue and again consumer will poll because time is 30 sec.

Now lets start polling.

A screenshot of a voting box

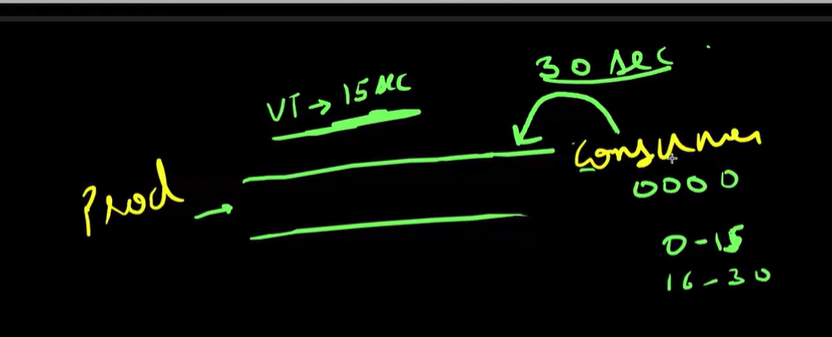
Description automatically generated

A screenshot of a computer

Description automatically generated

Here we can see our messages know and we can see receive count 2 here why?

Because consumer is not able to consume it in first 15 sec it went back to queue and again come.



Now if we refresh the page, it will go because consumer is not able to consume it.

To consume it we have to delete the messages. This is for only demo purpose all this should be done via code.

SES 🡪

Simple Email Service.

It is used for sending bulk email only. Like we are running a marketing campaign.

It doesn’t require subscribers. It is cheaper than SNS.

We can use images and ui design in SES but not in SNS.

It is the cheapest service in the market to send email in the bulk.

Now Let’s start with SES🡪

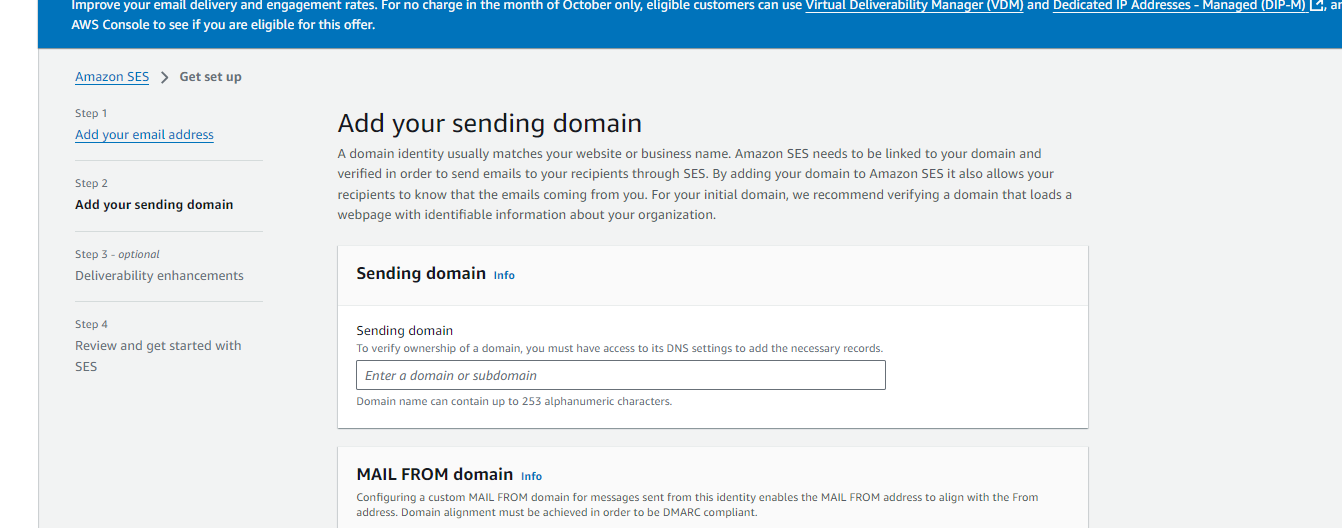
A screenshot of a computer

Description automatically generated

On clicking on get started a popup will open. Which is for setup your account.In first step you have to setup your email address

A screenshot of a computer

Description automatically generatedThis is one type of identity from which we send email , other is domain.



We have to purchase the domain for it , which we can skip for now. Review and get started. Now from the left menu we can check for the identities.

A screenshot of a computer

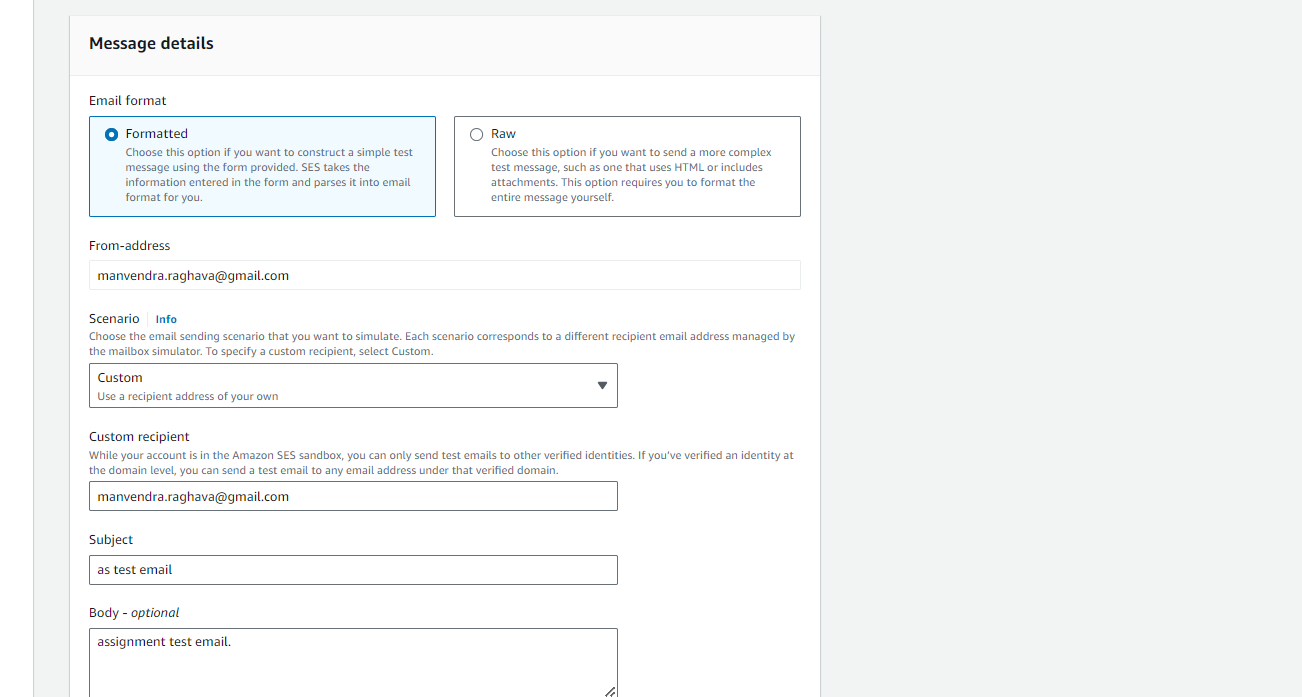
Description automatically generated

We have to verify the identity to send a email via domain or recipient email. Go in your email and verify.

A screenshot of a computer

Description automatically generated

We can see it verified , now send a test email. Select the user and click on test email.



Fill all the details and send the email.

A yellow rectangular object with black text

Description automatically generated